IDS 09/877,411

(19) Europäisches Patentamt
European Patent Office
Office europé n des brevets

(11) EP 0 996 052 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
26.04.2000 Bulletin 2000/17

(51) Int. Cl.[7]: **G06F 3/033**

(21) Application number: 99302869.5

(22) Date of filing: 13.04.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 19.10.1998 JP 29704698

(71) Applicant: FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)

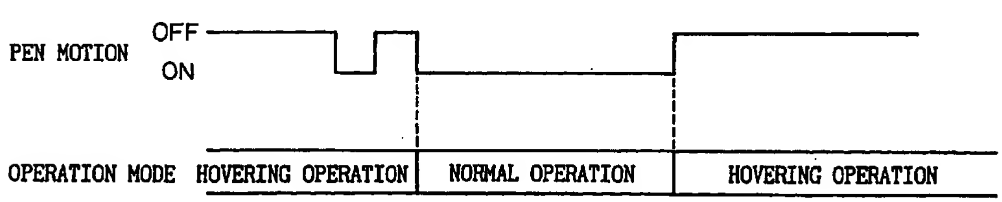
(72) Inventor: Kuribayashi, Tsuyoshi
Nakahara, Kawasaki, Kanagawa 211-8588 (JP)

(74) Representative:
Fenlon, Christine Lesley et al
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(54) **Input processing method and input control apparatus**

(57) An input processing method and input control apparatus that enable switching between normal operation and hovering operation to be accomplished with a very simple operation in an information processing apparatus that uses a touch input device such as a tablet or the like as an input device. During a period that an improved hovering mode according to the present invention is set active, the time interval between a pen up motion and a subsequent pen down motion is detected. If the time interval is equal to or longer than a predetermined length of time, pen down information to be sent to the OS is set OFF, thereby effecting the hovering operation. On the other hand, if the time interval is shorter than the predetermined length of time, the pen down information to be sent to the OS is set ON, thereby effecting the normal operation. Thus, in the improved hovering mode, with an extra tap of a pen, the immediately following pen operation works as if it were performed in normal mode.

Fig.17



EP 0 996 052 A2

Description

[0001] The present invention relates to a method for processing input from a touch input device such as a tablet or the like used as an input device, and an input control apparatus for implementing the method.

[0002] Portable information processing apparatuses, which are becoming popular, utilize touch input devices such as tablets or digitizers as input devices instead of mice. Generally, a tablet or digitizer is integrated with a display such as a liquid crystal display by being mounted as a transparent plate over the display or placed underneath the display, and is configured to detect the position of a stylus pen or the like on the display when the stylus pen or the like, as a pointing device, contacts the display screen.

[0003] Such input devices, generally called touch screens or touch panels, have been implemented in various ways, including one that uses a resistive film (pressure sensitive tablet) consisting of transparent electrodes bonded to the surface of a display (CRT, LCD, plasma display, etc.) and performs an input operation by touching the screen with a finger or a pen, and one that performs an input operation on the screen with the digitizer, mounted underneath the display, detecting the position of the input by detecting the magnetism being generated at the tip of the pen (when the pen is touched to the screen, magnetism is generated, and the position of the input is detected by the electromagnetic induction type digitizer mounted underneath the display). In addition to such pressure sensitive tablet and electromagnetic induction digitizer methods, various methods of input position detection have been considered, such as an ultrasonic surface acoustic wave touch panel method that uses ultrasonic waves.

[0004] A tablet, like a mouse, is also used to specify a position in a graphical input operation, select a menu or activate a software program by manipulating an icon, and so on.

[0005] A mouse contains mouse buttons, and a mouse operation to press and release a mouse button is called a mouse click, the mouse operation being classified as a single click, double click, etc. according to how many times the button is pressed in succession. By putting the mouse cursor on a designated icon and clicking the mouse on it, desired processing can be specified. In the case of a pen operation on a tablet, if an operation to touch the pen or the like to the tablet (called a pen down motion) and then lift it off the tablet (called a pen up motion) is called a tap, then moving the mouse cursor and single-clicking or double-clicking the mouse corresponds to performing a single tap or double tap on a designated icon. The pen down and pen up motions here embrace operations performed not only by a pen but also by a finger or the like.

[0006] In a graphical input operation, when a tablet 10 is touched with a pen 14 as shown in Figure 1 of the accompanying drawings, a dot is drawn on a display 12 at the position of the touch. The figure shows the tablet 10 and display 12 as a single integral unit. When the pen 14 is moved whilst holding it touched to the tablet 10, as shown in Figure 2 of the accompanying drawings, a line corresponding in its movement is drawn on the display 12.

[0007] In the case of a mouse, there is a mouse operation, called drag, in which the mouse is moved while keeping the mouse button pressed down; in a graphical input operation, a line is drawn when the mouse is dragged. On the other hand, when the mouse is moved without holding down a mouse button, the mouse cursor merely moves on the display.

[0008] In the case of a graphical input operation by a pen on a tablet, on the other hand, if a dot or line were always drawn with a pen down motion, an operation to move the cursor without drawing a dot or line, such as the operation shown in Figure 3 of the accompanying drawings, could not be accomplished. To avoid this, an operation mode is provided in which even when the pen is actually touched to the tablet, the motion is not interpreted as a pen down motion but the mouse cursor is merely caused to move. This mode is called a hovering mode. To switch the operation between normal mode and hovering mode, a normal mode selection button 16 and a hovering mode selection button 18 are provided outside the display screen, as shown in Figure 4 of the accompanying drawings. In an icon manipulation in the hovering mode, a pen tap on an icon only results in positioning the cursor on that icon.

[0009] In this way, an input operation on a touch input device such as a tablet or digitizer has required mode switching using the buttons provided outside the screen, and in the case of an input operation that requires frequent mode switching, the operation has been very tiring and inefficient for an operator.

[0010] It is desirable to provide an input processing method and input control apparatus for a touch input device that enables operation mode switching to be accomplished with a very simple operation.

[0011] According to an embodiment of a first aspect of the present invention, there is provided an input processing method for a device which provides input by performing a touch motion on an operating surface, comprising: a first step of detecting the length of time of a non-touch state in which no touch motion is performed on the operating surface; and a second step of determining information indicating a touch state in accordance with the detected length of time when a touch motion has occurred.

[0012] According to an embodiment of a second aspect of the present invention, there is provided an input processing method for a device which provides input by performing a touch motion on an operating surface, comprising: a first step of detecting the number of successive occurrences of the touch motion; and a second step of determining information indicating a touch state in accordance with the detected number of occur-

rences.

[0013] According to an embodiment of a third aspect of the present invention, there is provided an input processing method for a device which provides input by performing a touch motion on an operating surface, comprising: a first step of detecting the number of occurrences of the touch motion over a predetermined length of time; and a second step of determining information indicating a touch state in accordance with the detected number of occurrences.

[0014] According to an embodiment of a fourth aspect of the present invention, there is provided an input processing method for a device which provides input by performing a touch motion on an operating surface, comprising the steps of: detecting the number of successive occurrences of the touch motion; and determining a corresponding mouse operation in accordance with the detected number of occurrences.

[0015] Furthermore, embodiments of the present invention, can provide an input control apparatus for implementing the above method, and a program recording medium which is readable by a computer utilizing the input control apparatus, and on which is recorded a program for implementing the above method.

[0016] Reference will now be made, by way of example, to the accompanying drawings, in which:

Figure 1, discussed hereinbefore, is a diagram for explaining the motion for drawing a dot with a pen;

Figure 2, discussed hereinbefore, is a diagram for explaining the motion for drawing a line with a pen;

Figure 3, discussed hereinbefore, is a diagram for explaining the motion for merely moving cursor coordinates without drawing a line;

Figure 4, discussed hereinbefore, is a diagram for explaining mode switching buttons according to a previously-considered input device;

Figure 5 is a perspective view of a portable information processing apparatus to which an embodiment of the present invention can be applied;

Figures 6A and 6B are diagrams for use in explaining a pressure sensitive tablet and an electromagnetic induction tablet, respectively, as representative examples of tablets;

Figure 7 is a diagram showing the details of a hot pad;

Figure 8 is a block diagram showing the system configuration of the portable information processing apparatus shown in Figure 5;

Figure 9 is a flowchart illustrating the sequence of processing performed by an exemplary tablet controller;

Figure 10 is a diagram showing the possible format of data sent out from a tablet controller;

Figure 11 is a diagram for use in explaining the possible operation of a tablet controller;

Figure 12 is a flowchart illustrating a pen driver processing sequence;

Figure 13 is a diagram for use in explaining the possible operation of a pen driver;

Figure 14 is a flowchart illustrating a processing sequence of an OS during graphical input processing;

Figures 15A, 15B, and 15C are diagrams for use in explaining a method of operation mode switching for use in an embodiment of the present invention;

Figures 16A, 16B, 16C, and 16D are diagrams for use in explaining a method of operation mode switching used in a previously-considered input device;

Figure 17 is a diagram for explaining pen motion and operation modes according to an embodiment of the present invention;

Figure 18 is a diagram for explaining pen motion and operation modes according to a previously-considered input device;

Figure 19 is a diagram showing an example of an icon manipulation screen; and

Figures 20A and 20B are flowcharts illustrating a processing sequence of an OS during an icon manipulation.

[0017] An embodiment of the present invention will be described below with reference to the accompanying drawings. The embodiment hereinafter described deals with an example that uses a pen (or stylus), but the invention is also applicable to devices that are operated using fingers. For example, ultrasonic surface acoustic wave touch panels and pressure sensitive tablets permit position input by touching them with a finger without using a pen. Embodiments of the present invention do not exclude application to such devices.

[0018] Figure 5 is a perspective view of a portable information processing apparatus to which an embodiment of the present invention can be applied. A tablet 10 and a liquid crystal display 12 are combined together with one overlaid on top of the other, and function as electronic "paper". Various types of tablet 10 are commercially implemented, and embodiments of the present invention are applicable to any type of tablet.

[0019] Figures 6A and 6B are diagrams for explaining a pressure sensitive tablet and an electromagnetic induction tablet, respectively, as representative examples of tablets. The pressure sensitive tablet 10 of Figure 6A consists of a transparent film and glass, and is mounted over the liquid crystal display 12. When the film is brought into contact with the glass by being pressed with a pen or the like, there occurs a change in the resistance value, based on which the position where the contact is made is detected. The electromagnetic induction tablet 10 of Figure 6B is constructed from a sensor plate containing a sensor coil, and is placed underneath the liquid crystal display 12. Pen position is determined by detecting the magnetism, generated from the tip of the pen, by means of the sensor coil.

[0020] The details of a hot pad 20 in Figure 5 are

illustrated in Figure 7. A hovering icon 22 on the hot pad 20 is an icon used for switching operation between the normal mode and an improved hovering mode according to the present invention. When the hovering icon 22 is touched during operation in the normal mode, the tablet 10 and display 12 operate in the improved hovering mode according to the present invention until this icon is touched again.

[0021] Figure 8 is a block diagram showing the system configuration of the portable information processing apparatus (the so-called pen computer) shown in Figure 5. A multichip module (MCM) 32, south bridge 38, and video graphics array (VGA) controller 36 are connected to a PCI (Peripheral Component Interconnect) bus 30. The south bridge 38, tablet controller 52, and ROM (read-only memory) 54 are connected to an ISA (Industry Standard Architecture) bus 50.

[0022] The multichip module 32 contains multiple bare silicon chips mounted on a small printed board, and comprises a CPU, a north bridge, and a secondary cache memory. The north bridge is a PCI system controller for controlling data transfers between the CPU, main memory, secondary cache memory, PCI bus, etc. A synchronous DRAM (SDRAM) 34 as the main memory is connected to the multichip module 32 via a memory bus. The SDRAM 34 is characterized by its operation synchronized to the system bus clock.

[0023] The VGA controller 36 controls the liquid crystal display 12 under the VGA standard. The south bridge 38 is a bridge, mounted with the PCI bus, that provides connections between the PCI bus and ISA bus on the motherboard, and controls data transfers between the CPU, memory, input/output devices, etc. A hard disk drive (HDD) 40 as an auxiliary storage device is connected to the south bridge 38.

[0024] The tablet controller 52 is responsible for controlling the processing of input from the tablet 10 and detecting an operation performed on the hot pad 20, and includes a CPU 52a, a memory (RAM) 52b, and an interface circuit 52c. The ROM 54 holds therein an initial program loader.

[0025] A port replicator 60, also called an expansion I/O box, connector box, expansion unit, or the like, is an adaptor which is connected to a notebook computer or a pen computer to enhance their expandability. Usually, it accommodates various connectors, such as a serial port 60d, a parallel port 60a, and a CRT connector 60f, as well as connection ports for a keyboard 60b, a mouse 60c, a floppy disk drive 60e, an Ethernet port, etc. Connectors that are infrequently used when the notebook computer or pen computer is taken on the road are accommodated in the port replicator.

[0026] In the present embodiment, the floppy disk drive 60e is connected via the port replicator 60, but the floppy disk drive may be connected directly to the pen computer, or alternatively, the pen computer itself may be constructed to contain a floppy disk. Further, the floppy disk drive may be incorporated in a docking station, rather than being connected via the port replicator. A docking station is a device that contains a SCSI card, an Ethernet card, a CD-ROM drive, a floppy disk drive, an additional PC card expansion bus, etc., and mostly is shaped so that a notebook computer can fit into it.

[0027] Programs may be provided using the CD-ROM drive or the floppy disk drive accommodated in the docking station.

[0028] Figure 9 is a flowchart illustrating the sequence of processing that the CPU 52a within the tablet controller 52 performs in accordance with the program loaded into the memory 52b of the controller 52. This program is usually stored on the HDD 40, and is loaded into the memory of the tablet controller 52 at system power-up. The program may also be provided on other recording media such as a floppy disk or a CD-ROM. The illustrated process is invoked as an interrupt service routine when a pen down motion is performed on the tablet 10.

[0029] First, in step 102, upon detecting the pen down motion, a pen down state flag and a pen down state start flag are both set to 1, and a pen down state end flag is cleared to 0. The pen down state flag is a flag that is set to indicate that the current state is the pen down state, and the pen down state start flag is a flag that is set to 1 only once when a transition is made to the pen down state, while the pen down state end flag is a flag that is set to indicate the end of the pen down state. These flags are implemented in the memory 52b within the tablet controller 52. Next, in step 104, it is determined whether the current operation mode is the improved hovering mode or not, based on the operation performed on the hovering icon 22 on the hot pad 20. If the current mode is not the improved hovering mode, that is, if the current mode is the normal mode, the process proceeds to step 106; on the other hand, if the current mode is the improved hovering mode, the process proceeds to step 108.

[0030] In step 106, an improved hovering mode flag, a flag that is set to indicate the improved hovering mode, is cleared to 0, i.e., set OFF. On the other hand, in step 108, the improved hovering mode flag is set to 1, i.e., ON. The improved hovering mode flag also is implemented in the memory 52b within the tablet controller 52. In step 110, which follows step 106 or 108, coordinate data indicating the position at which the tablet 10 is pressed by the pen is detected. In the case of a pressure sensitive tablet, the coordinate data is detected by applying a voltage and measuring the resistance.

[0031] Next, in step 112, transmission data consisting of status (1 byte) and coordinate data (4 bytes), as shown in Figure 10, is assembled in the memory 52b within the tablet controller 52, and is sent to the pen driver described later. The status byte contains a pen down state bit (pen down information), a pen down state start bit, a pen down state end bit, and an improved hovering mode bit, respectively indicating the values of the pen down state flag, pen down state start flag, pen

down state end flag, and improved hovering mode flag.

[0032] In step 114, it is determined ~~whether the current state~~ is the pen up state, ~~and if the state is not the~~ pen up state, that is, if the state is still the pen down state, the process proceeds to step 116; on the other hand, if the state is the pen up state, the process proceeds to step 118. In step 116, the pen down state start flag is cleared to 0, and the process loops back to step 110. In step 118, on the other hand, the pen down state flag and the pen down state start flag are both cleared to 0, while setting the pen down state end flag to 1, and the status reflecting the contents of these flags is created and sent out. In this way, during the period from the pen down to the pen up, data is sent out at predetermined intervals of time (for example, every 5 ms).

[0033] The state transitions of the flags in the improved hovering mode in the above-described process will be described with reference to Figure 11. First, when the pen down motion is detected, the pen down state flag is set to 1, the pen down state start flag is set to 1, the pen down state end flag is cleared to 0, and the improved hovering mode flag is set to 1 in steps 102, 104, and 108, as shown in case (A) in Figure 11. Then, these flag states are written in the status bits and sent out in step 112 together with the coordinate data detected in step 110.

[0034] As long as the pen down state continues, the process consisting of steps 114, 116, 110, and 112 is carried out in a loop fashion, as shown in case (B) in Figure 11; during that period, the pen down state flag remains at 1, the pen down state start flag remains at 0, and the pen down state end flag remains at 0. During the loop process, the coordinate data is constantly updated in step 110, and the coordinate data and the status are sent out in step 112 at predetermined intervals of time.

[0035] When a pen up motion is detected to end the pen down state, the process proceeds from step 114 to step 118, and the pen down state flag and pen down state start flag are now at 0 and the pen down state end flag is set to 1, as shown in case (C) in Figure 11. This forms the final status to be sent out. In the above process, if a pen up motion is detected immediately following the pen down motion, the flags make transitions from the states shown in case (A) in Figure 11 directly to the states shown in case (C) in Figure 11 without passing through the states shown in case (B) in Figure 11.

[0036] Figure 12 is a flowchart illustrating a pen driver processing sequence carried out by the CPU 32a within the multichip module 32. This pen driver program is usually stored on the HDD 40, and is loaded into the SDRAM 34 at system power-up and executed on the SDRAM 34. The program may be provided on other storage media such as a floppy disk or a CD-ROM. The illustrated process is initiated upon ~~detecting~~ the arrival of data sent from the tablet controller 52. First, in step 202, the data (see Figure 10) sent from the tablet controller 52 is rec ived, and stored in a designated area within the SDRAM 34. Next, in step 204, the status in the received data is examined to determine whether the improved hovering mode bit is 1 or 0, and if the improved hovering mode bit is 1, the process proceeds to step 206; on the other hand, if the improved hovering mode bit is 0, the process proceeds to step 224.

[0037] In step 206, it is determined whether the pen down state end bit contained in the status byte in the received data is 1 or 0, and if the pen down state end bit is 1, the process proceeds to step 208; on the other hand, if the pen down state end bit is 0, the process proceeds to step 210. In step 208, the countdown timer is set and started, whereupon the routine is terminated. This countdown timer is provided to measure the time interval between a pen up motion and a pen down motion, thereby to determine whether or not the length of time of the pen up state is less than a predetermined threshold value. This timer is implemented as a software timer in the SDRAM 34.

[0038] In step 210, it is determined whether the pen down state start bit contained in the status is 1 or 0, and it the pen down state start bit is 1, the process proceeds to step 212; on the other hand, if the pen down state start bit is 0, the process proceeds to step 220. In step 212, the countdown timer is checked to determine whether the timer has counted down 0 or not, and if the timer has not yet counted down to 0, that is, if the length of time of the pen up state is shorter than the predetermined length of time, the process proceeds to step 214; on the other hand, if the timer has already counted down to 0, that is, if the length of time of the pen up state is longer than the predetermined length of time, the process proceeds to step 218.

[0039] In step 214, hovering valid flag is set to 1, and the process proceeds to step 216. This hovering valid flag is a flag that is set to indicate that the hovering operation is to be performed validly in the improved hovering mode as it is originally intended. The flag is implemented in the SDRAM 34. In step 216, the pen down state bit in the status is changed to 0, and the process proceeds to step 224. On the other hand, in step 218, the hovering valid flag is cleared to 0 to invalidate the hovering operation in the improved hovering mode as an exception and thereby to effect the normal mode; then, the process proceeds to step 224.

[0040] In step 220, which is carried out when the pen down state start bit is detected as being 0 in step 210, the hovering valid flag is checked to determine whether it is at 1 or 0, and if the hovering valid flag is at 1, the process proceeds to step 222; on the other hand, if the hovering valid flag is at 0, the process proceeds to step 224. In step 222, the pen down state bit in the status is changed to 0, and the process proceeds to step 224. In the final step 224, the coordinate data and the status containing the pen down state bit after completion of the above processing are sent to the operating system (OS).

[0041] As the result of th above pen driver

processing, the pen down state bit (pen down information) in the status reported to the OS along with the coordinate data is as follows. When the mode is not the improved hovering mode, that is, when the mode is the normal mode, steps 202, 204, and 224 are carried out in sequence, so that the pen down state bit remains unchanged at 1 and is reported as is.

[0042] Pen driver operation in the improved hovering mode, on the other hand, will be described below with reference to Figure 13. In the improved hovering mode, the operation differs depending on the time interval between the pen up motion and the pen down motion detected in the above process, that is, depending on whether or not the length of time of the pen up state is less than the threshold value set by the countdown timer.

[0043] First, since the pen down state end bit is 1 in the final status, that is, the status received when the pen down state has ended by detecting a pen up motion, the steps 202, 204, 206, and 208 are carried out, as shown in case (A) in Figure 13, and the countdown timer is thus set and started to count the time until the next pen down motion is detected.

[0044] Then, when the next pen down occurs with the countdown timer already reaching 0, that is, when a pen down motion is detected after the predetermined time has elapsed from the preceding pen up motion, the first status received indicates that the pen down state bit is 1, the pen down state start bit is 1, and the pen down state end bit is 0, as shown in case (B) in Figure 13, and at this time, the countdown timer is already at 0; accordingly, the steps 202, 204, 206, 210, 212, 214, 216, and 224 are carried out in sequence. In the process, the hovering valid flag is set to 1 in step 214. The pen down state bit is changed to 0 in step 216, and the change state bit is reported to the OS in step 224. In other words, the number of pen down motions performed within the predetermined time is detected using the countdown timer, and if it is determined in step 212 that the countdown timer reads 0, that means that the number of pen down motions (touch motions) occurring in succession within the predetermined time is 1; as a result, the pen down state bit is not set ON, and it is determined that the state is not the pen down state.

[0045] In intermediate status received during the pen down state following the states shown in case (B) in Figure 13, the pen down state bit is 1, the pen down state start bit is 0, and the pen down end bit is 0, as shown in case (C) in Figure 13 and, at this time, the hovering valid flag is at 1; accordingly, the steps 202, 204, 206, 210, 220, 222, and 224 are carried out in sequence. Here again, the pen down state bit is changed to 0 in step 222, and the changed state bit is reported to the OS in step 224.

[0046] On the other hand, when the next pen down occurs with the countdown timer yet to reach 0, that is, when a pen down motion is detected befor the predetermined time has elapsed from the preceding pen up motion, the first status received indicates that th pen down state bit is 1, the pen down state start bit is 1, and the pen down state end bit is 0, as shown in case (D) in Figure 13, and at this time, the countdown timer has not yet reached 0; accordingly, the steps 202, 204, 206, 210, 212, 218, and 224 are carried out in sequence. In the process, the hovering valid flag is cleared to 0 in step 218. The pen down state bit remains unchanged at 1 and is reported to the OS as it is. In other words, the number of pen down motions performed within the predetermined time is detected using the countdown timer, and if it is determined in step 212 that the countdown timer has not yet reached 0, that means that the number of pen down motions (touch motions) occurring in succession within the predetermined time is 2 or more; as a result, the pen down state bit is set ON accordingly, and it is determined that the state is the pen down state.

[0047] In intermediate status received during the pen down state following the states shown in case (D) in Figure 13, the pen down state bit is 1, the pen down state start bit is 0, and the pen down state end bit is 0, as shown in case (E) in Figure 13, and at this time, the hovering valid flag is at 0; accordingly, the steps 202, 204, 206, 210, 220, and 224 are carried out in sequence. As a result, the pen down state bit remains unchanged at 1, and is reported to the OS as it is.

[0048] Following the states shown in case (B), (C), (D), or (E) in Figure 13, when the pen down states ends by detecting a pen up motion, and the final status is received, the countdown timer is started, as previously described, to prepare for the next pen down, as shown in case (A) in Figure 13.

[0049] Figure 14 is a flowchart illustrating the processing sequence of the OS during graphical input processing. This OS program is usually stored on the HDD 40, and is loaded into the SDRAM 34 at system power-up and executed on the SDRAM 34 by the CPU 32a within the multichip module 32. This OS program may be provided on other storage media such as a floppy disk or a CD-ROM. The illustrated process is initiated upon arrival of the data sent from the pen driver. First, in step 302, the data sent from the pen driver is received, and stored in a designated area within the SDRAM 34. Next, in step 304, it is determined whether the pen down state bit in the status is 1 or 0, and if the pen down state bit is 1, the process proceeds to step 306; on the other hand, if the pen down state bit is 0, the process proceeds to step 308.

[0050] In step 306, the cursor is moved while maintaining the pen down state which is equivalent to the state in which a mouse is clicked and held down. This causes a dot or line to be drawn, just as if a mouse was dragged. On the other hand, in step 308, the cursor is merely moved to the position indicated by the received coordinate data.

[0051] According to the above-described tablet controller, pen driver, and OS processing operations, by just performing an extra pen tap in the improved hover-

ing mode, the immediately following pen tap works just like a pen tap performed in the normal mode. That is, supposing that the state shown in Figure 15A is a state in the improved hovering mode, if a line drawing motion is performed on the tablet after performing a pen tap once, as shown in Figure 15B, a line is drawn on the display as in the normal mode, as shown in Figure 15C. After a series of drawing motions, a pen up state is entered, and after the elapse of a certain amount of time, the hovering operation is again enabled. Using the previously-considered input device, the pen had to be moved outside the screen to switch the mode, as shown in Figures 16A, 16B, 16C, and 16D, but using an embodiment of the present invention, a smooth operation becomes possible.

[0052] Pen motion and operation modes will be described in more detail when a line is drawn by temporarily entering the normal operation mode during the improved hovering mode and then resuming the hovering operation. In an embodiment of the present invention, immediately before drawing a line, a pen tap is performed to switch the operation, and immediately after that, a line is drawn; when the pen is lifted off the tablet after drawing the line, the hovering operation automatically resumes, as shown in Figure 17. In the previously-considered method, on the other hand, at timing A before drawing a line, the appropriate button had to be pressed to switch to the normal mode, as shown in Figure 18, and then at timing B after drawing the line, another button had to be pressed to switch back to the hovering mode. An embodiment of the present invention can eliminate the need for such troublesome button operations.

[0053] Next, a description will be given of icon manipulations in the improved hovering mode according to an embodiment of the present invention. Suppose an icon manipulation screen such as shown in Figure 19 is used in the improved hovering mode. In this case also, by just performing an extra pen tap, the immediately following pen tap works just like a pen tap performed in the normal mode. More specifically, since the first tap is used for operation switching, the equivalent of a single click of a mouse is accomplished by a double tap of the pen, and the equivalent of a double click of a mouse is accomplished by a triple tap of the pen.

[0054] Figures 20A and 20B are flowcharts illustrating the processing sequence of the OS during an icon manipulation. This OS program is usually stored on the HDD 40, and is loaded into the SDRAM 34 at system power-up and executed on the SDRAM 34 by the CPU 32a within the multichip module 32. This OS program may be provided on other storage media such as a floppy disk or a CD-ROM. The illustrated process is initiated upon arrival of the data sent from the pen driver. As earlier described, for successively occurring pen taps, the pen driver sets the pen down state bit to 1 for the second and later taps, and sends this bit state to the OS. Accordingly, the OS interprets the second tap as the first mouse click.

[0055] First, in step 402, the data sent from the pen driver is received, and stored in a designated area within the SDRAM 34. Next, in step 404, it is determined whether the pen down state bit and pen down state start bit in the status are both 1 or not, and if the result is YES, the process proceeds to step 406; on the other hand, if the result is NO, the routine is terminated.

[0056] In step 406, it is determined whether an interrupt timer value, which will be set in the following step 408, is 0 or not, and if the timer value is 0, the process proceeds to step 408; on the other hand, if the timer value is not 0, the process proceeds to step 410. In step 408, the interrupt timer is set, and the routine is terminated. In step 410, on the other hand, it is decided that the next pen down has occurred within the predetermined time from the previous pen down, and the interrupt timer is reset. In the next step 412, it is determined that a double click has occurred.

[0057] When an interrupt is caused by the interrupt timer set in step 408, the process shown in Figure 20B is carried out. Since this process is invoked when the next pen down did not occur within the predetermined time from the previous pen down, it is determined in step 450 that a single click has occurred.

[0058] When a single click is detected, a file selection or other operation is performed, and when a double click is detected, a program in the file is activated, for example.

[0059] As described above, according to an embodiment of the present invention, in an information processing apparatus that uses a touch input device such as a tablet or digitizer instead of a mouse as an input device, switching between normal operation and hovering operation can be accomplished with a very simple operation, enhancing its usability. Though the embodiment has been described by dealing with an example that uses a pen (or stylus), an embodiment of the present invention is also applicable to an apparatus that is operated by a finger.

[0060] The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiment is therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

[0061] It will be appreciated that computer programs can be stored on a variety of media, such as floppy disks, hard disks, CD-ROMs, etc., or downloaded into a computer, for example from the Internet. It is intended that the attached claims should encompass computer programs in all these, and similar, forms.

Claims

1. An input processing method for a device which provides input by performing a touch motion on an operating surface, comprising:

   a first step of detecting the length of time of a non-touch state in which no touch motion is performed on said operating surface; and
   a second step of determining information indicating a touch state in accordance with said detected length of time when a touch motion has occurred.

2. An input processing method as claimed in claim 1, wherein when a touch state occurs, said first step detects the length of time of said non-touch state that follows said touch state.

3. An input processing method as claimed in claim 1, wherein said second step determines that the state is the touch state if said detected length of time is less than a predetermined length of time.

4. An input processing method for a device which provides input by performing a touch motion on an operating surface, comprising:

   a first step of detecting the number of successive occurrences of said touch motion; and
   a second step of determining information indicating a touch state in accordance with said detected number of occurrences.

5. An input processing method as claimed in claim 4, wherein said first step detects the number of occurrences of said touch motion over a predetermined length of time.

6. An input processing method for a device which provides input by performing a touch motion on an operating surface, comprising:

   a first step of detecting the number of occurrences of said touch motion over a predetermined length of time; and
   a second step of determining information indicating a touch state in accordance with said detected number of occurrences.

7. An input processing method as claimed in any one of claims 4 to 6, wherein when said number of occurrences is 1, said second step determines that the state is not the touch state occurring in relation to said touch motion, and when said detected number of occurrences is 2 or more, then determines that the state is the touch state.

8. An input processing method as claimed in any one of claims 4 to 6, wherein when said detected number of occurrences is 2, it is determined that a single click has occurred, and/or when said detected number of occurrences is 3, it is determined that a double click has occurred.

9. An input processing method as claimed in any one of claims 1 to 7, wherein

   said device is a device that displays a cursor in a display section,
   said cursor has a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and
   there is included a step in which said cursor changes from said second state to said first state in response to said touch state indicating information.

10. An input processing method as claimed in any one of claims 1 to 9, including the step of determining information indicating the termination of said touch state after said second step has determined said touch state indicating information and said touch state has ended.

11. An input processing method as claimed in any one of claims 1 to 7, wherein

    said device is a device that displays a cursor in a display section,
    said cursor has a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and
    there is included a step in which said cursor changes from said first state to said second state when said touch motion has ended.

12. An input processing method as claimed in any one of claims 1 to 11, including the step of performing information processing in response to said touch state indicating information.

13. An input processing method as claimed in any one of claims 1 to 4, wherein said touch motion is performed using a pen on said operating surface, and

    said non-touch state is a pen up state, and/or said touch state is a pen down state.

14. An input processing method as claimed in claim 9, wherein said first state is a cursor-clicked state, and/or said second stat is a hovering state.

15. An input processing method for a device which provides input by performing a touch motion on an operating surface, comprising the steps of:

detecting the number of successive occurrences of said touch motion; and
determining a corresponding mouse operation in accordance with said detected number of occurrences.

16. An input processing method as claimed in claim 15, wherein said corresponding mouse operation determining step determines that a single click has occurred when said detected number of occurrences is 2, and/or determines that a double click has occurred when said detected number of occurrences is 3.

17. An input processing method as claimed in any one of claims 1 to 16, wherein said device comprises a display section and said touch motion operating surface is arranged in said display section.

18. An input processing method as claimed in any one of claims 1 to 16, wherein each of said steps is carried out only when a designated mode is set active.

19. An input control apparatus for a touch input device which provides input by performing a touch motion on an operating surface, comprising:

a first unit detecting the length of time of a non-touch state in which no touch motion is performed on said operating surface; and
a second unit determining information indicating a touch state in accordance with said detected length of time when a touch motion has occurred.

20. An input control apparatus as claimed in claim 19, wherein when a touch state occurs, said first step detects the length of time of said non-touch state that follows said touch state.

21. An input control apparatus as claimed in claim 19, wherein said second unit determines that the state is the touch state if said detected length of time is less than a predetermined length of time

22. An input control apparatus for a device which provides input by performing a touch motion on an operating surface, comprising:

a first unit detecting the number of successive occurrences of said touch motion; and
a second unit determining information indicating a touch state in accordance with said detected number of occurrences.

23. An input control apparatus as claimed in claim 22, wherein said first unit detects the number of occurrences of said touch motion over a predetermined length of time.

24. An input control apparatus for a device which provides input by performing a touch motion on an operating surface, comprising:

a first unit detecting the number of occurrences of said touch motion over a predetermined length of time; and
a second unit determining information indicating a touch state in accordance with said detected number of occurrences.

25. An input control apparatus as claimed in any one of claims 22 to 24, wherein when said number of occurrences is 1, said second unit determines that the state is not the touch state occurring in relation to said touch motion, and when said detected number of occurrences is 2 or more, then determines that the state is the touch state.

26. An input control apparatus as claimed in any one of claims 22 to 25, wherein when said detected number of occurrences is 2, it is determined that a single click has occurred, and/or when said detected number of occurrences is 3, it is determined that a double click has occurred.

27. An input control apparatus as claimed in any one of claims 19 to 25, wherein

said device is a device that displays a cursor in a display section,
said cursor has a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and
there is included a unit for changing said cursor from said second state to said first state in response to said touch state indicating information.

28. An input control apparatus as claimed in any one of claims 19 to 27, including a unit determining information indicating the termination of said touch state after said second unit has determined said touch state indicating information and said touch state has ended.

29. An input control apparatus as claimed in any one of claims 19 to 25, wherein

said device is a device that displays a cursor in a display section,

said cursor has a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and

there is included a unit changing said cursor from said first state to said second state when said touch motion has ended.

30. An input control apparatus as claimed in any one of claims 19 to 29, including a unit performing information processing in response to said touch state indicating information.

31. An input control apparatus as claimed in any one of claims 19 to 22, wherein said touch motion is performed using a pen on said operating surface, and

said non-touch state is a pen up state, and/or said touch state is a pen down state.

32. An input control apparatus as claimed in claim 27, wherein said first state is a cursor-clicked state, and/or said second state is a hovering state.

33. An input control apparatus for a device which provides input by performing a touch motion on an operating surface, comprising:

a unit detecting the number of successive occurrences of said touch motion; and
a unit determining a corresponding mouse operation in accordance with said detected number of occurrences.

34. An input control apparatus as claimed in claim 33, wherein said corresponding mouse operation determining unit determines that a single click has occurred when said detected number of occurrences is 2, and/or determines that a double click has occurred when said detected number of occurrences is 3.

35. An input control apparatus as claimed in any one of claims 19 to 33, wherein each of said units operates only when a designated mode is set active.

36. An information processing apparatus comprising: said touch input device; and an input control apparatus as claimed in any one of claims 19 to 34.

37. An information processing apparatus comprising: said touch input device; a display device; and an input control apparatus as claimed in any one of claims 19 to 34, and wherein: the operating surface of said touch input device is arranged in said display device.

38. A recording medium readable by a computer, said computer using a device which provides input by performing a touch motion on an operating surface, said recording medium having a program recorded thereon for causing said computer to implement:

a first function detecting the length of time of a non-touch state in which no touch motion is performed on said operating surface; and
a second function determining information indicating a touch state in accordance with said detected length of time when a touch motion has occurred.

39. A recording medium as claimed in claim 38, wherein when a touch state occurs, said first function detects the length of time of said non-touch state that follows said touch state.

40. A recording medium as claimed in claim 38, wherein said second function determines that the state is the touch state if said detected length of time is less than a predetermined length of time.

41. A recording medium readable by a computer, said computer using a device which provides input by performing a touch motion on an operating surface, said recording medium having a program recorded thereon for causing said computer to implement:

a first function detecting the number of successive occurrences of said touch motion; and
a second function for determining information indicating a touch state in accordance with said detected number of occurrences.

42. A recording medium as claimed in claim 41, wherein said first function detects the number of occurrences of said touch motion over a predetermined length of time.

43. A recording medium readable by a computer, said computer using a device which provides input by performing a touch motion on an operating surface, said recording medium having a program recorded thereon for causing said computer to implement:

a first function detecting the number of occurrences of said touch motion over a predetermined length of time; and
a second function determining information indicating a touch state in accordance with said detected number of occurrences.

44. A recording medium as claimed in any one of claims 41 to 43, wherein when said number of occurrences is 1, said second function determines that the state is not the touch state occurring in rela-

tion to said touch motion, and when said detected number of occurrences is 2 or more, then determines that the stat is the touch state.

45. A recording medium as claimed in any one of claims 41 to 44, including a function for causing said computer to determine that a single click has occurred when said detected number of occurrences is 2, and/or a function for causing said computer to determine that a double click has occurred when said detected number of occurrences is 3.

46. A recording medium as claimed in any one of claims 38 to 44, wherein

said device is a device that displays a cursor in a display section, said cursor having a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and
there is included a function changing said cursor from said second state to said first state in response to said touch state indicating information.

47. A recording medium as claimed in any one of claims 38 to 46, including a function for determining information indicating the termination of said touch state after said second function has determined said touch state indicating information and said touch state has ended.

48. A recording medium as claimed in any one of claims 38 to 44, wherein

said device is a device that displays a cursor in a display section,
said cursor has a first state for directing the processing of a manipulation target displayed in said display section, and a second state for not directing the processing of said manipulation target, and
there is included a function for changing said cursor from said first state to said second state when said touch motion has ended.

49. A recording medium as claimed in any one of claims 38 to 48, including a function for performing information processing in response to said touch state indicating information.

50. A recording medium as claimed in any one of claims 38 to 41, wherein said touch motion is performed using a pen on said operating surface, and

said non-touch state is a pen up state, and/or said touch state is a pen down state.

51. A recording medium as claimed in claim 46, wherein said first state is a cursor-clicked state, and/or said second state is a hovering state.

52. A recording medium readable by a computer, said computer using a device which provides input by performing a touch motion on an operating surface, said recording medium having a program recorded thereon for causing said computer to implement:

a function detecting the number of successive occurrences of said touch motion; and
a function determining a corresponding mouse operation in accordance with said detected number of occurrences.

53. A recording medium as claimed in claim 52, wherein said corresponding mouse operation determining function determines that a single click has occurred when said detected number of occurrences is 2, and/or determines that a double click has occurred when said detected number of occurrences is 3.

54. A recording medium as claimed in any one of claims 38 to 53, wherein said device comprises a display section and said touch motion operating surface is arranged in said display section.

55. A computer program which, when running on a computer, is operable to carry out the steps of a method as claimed in any one of claims 1 to 18.

Fig.1

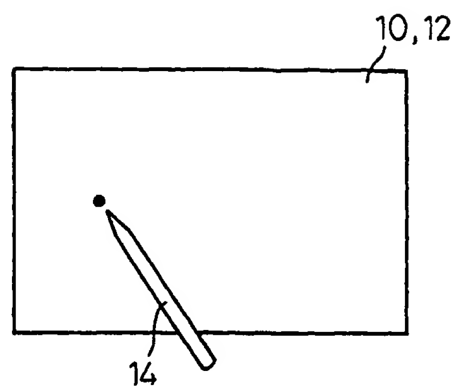


Fig.2

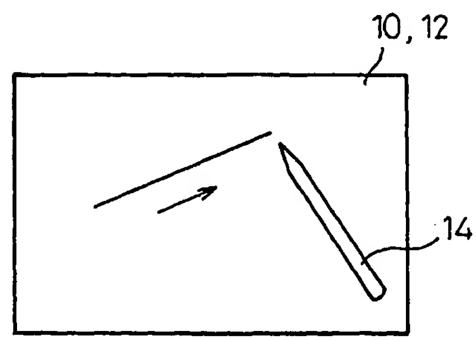


Fig.3

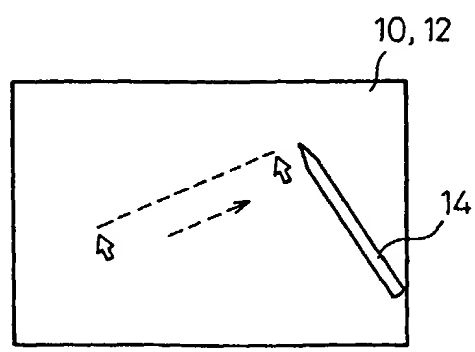


Fig.4

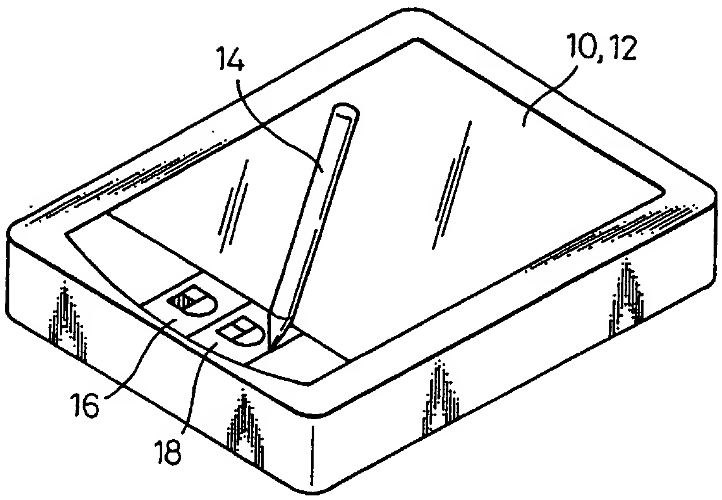


Fig.5

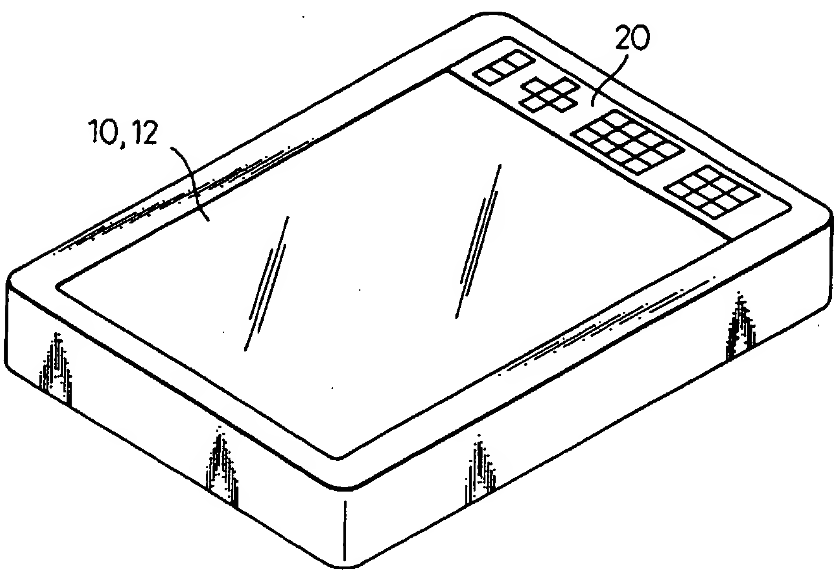


Fig.6A

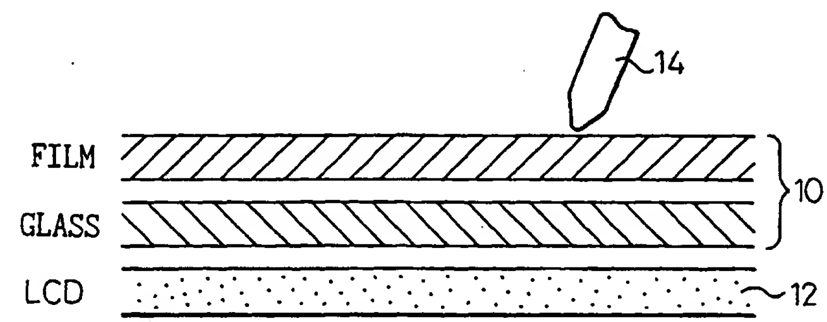


Fig.6B

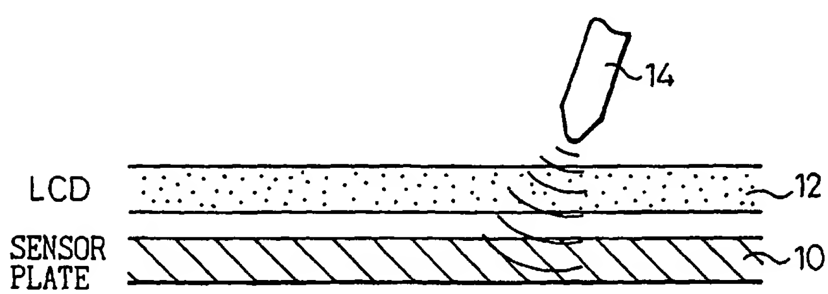


Fig.7

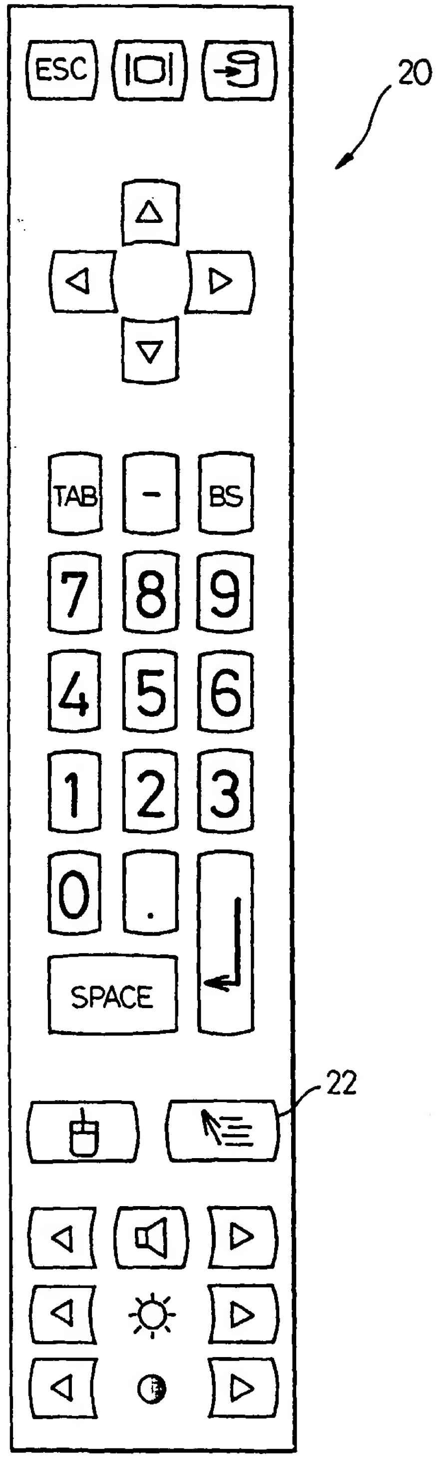


Fig.8

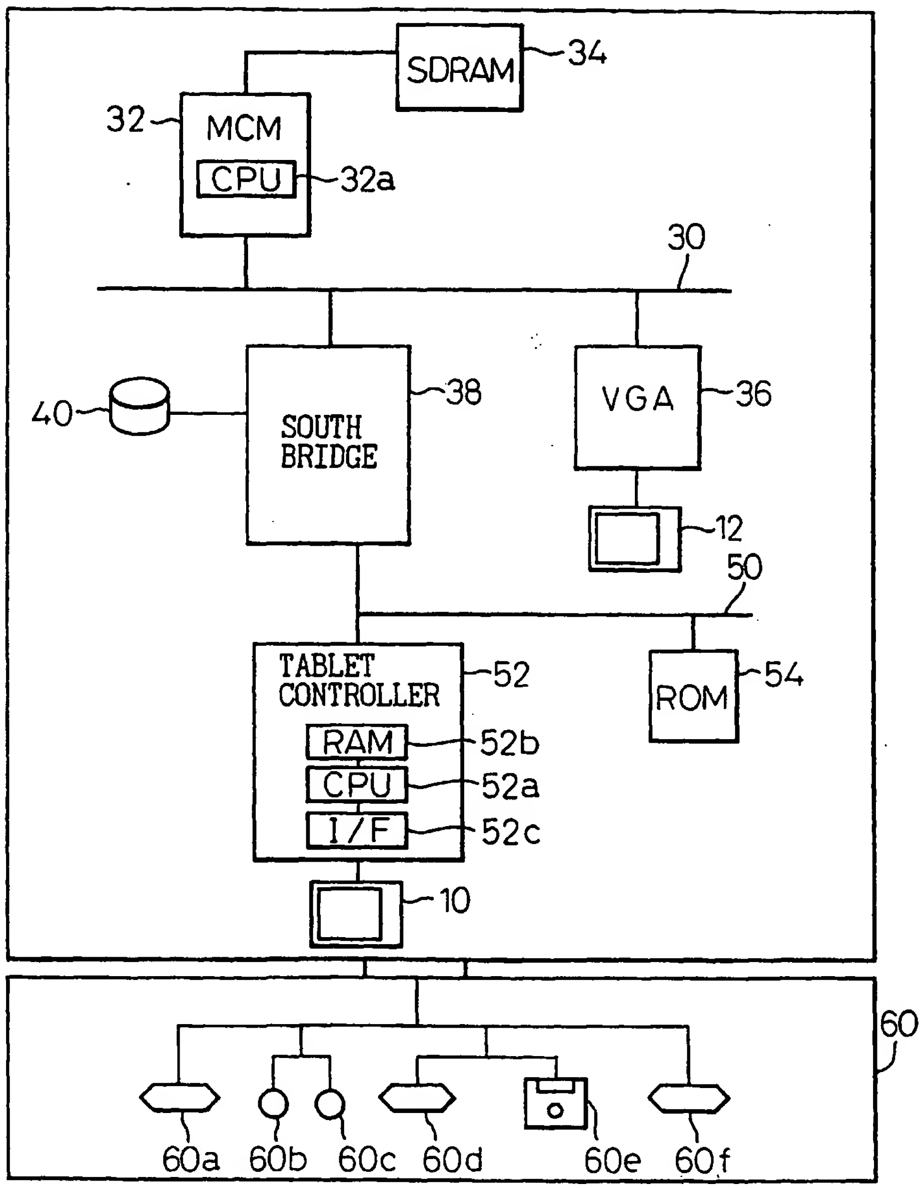

SDRAM
34
32
MCM
CPU
32a
30
40
SOUTH BRIDGE
38
VGA
36
12
50
TABLET CONTROLLER
52
ROM
54
RAM
52b
CPU
52a
I/F
52c
10
60
60a
60b
60c
60d
60e
60f

Fig.9

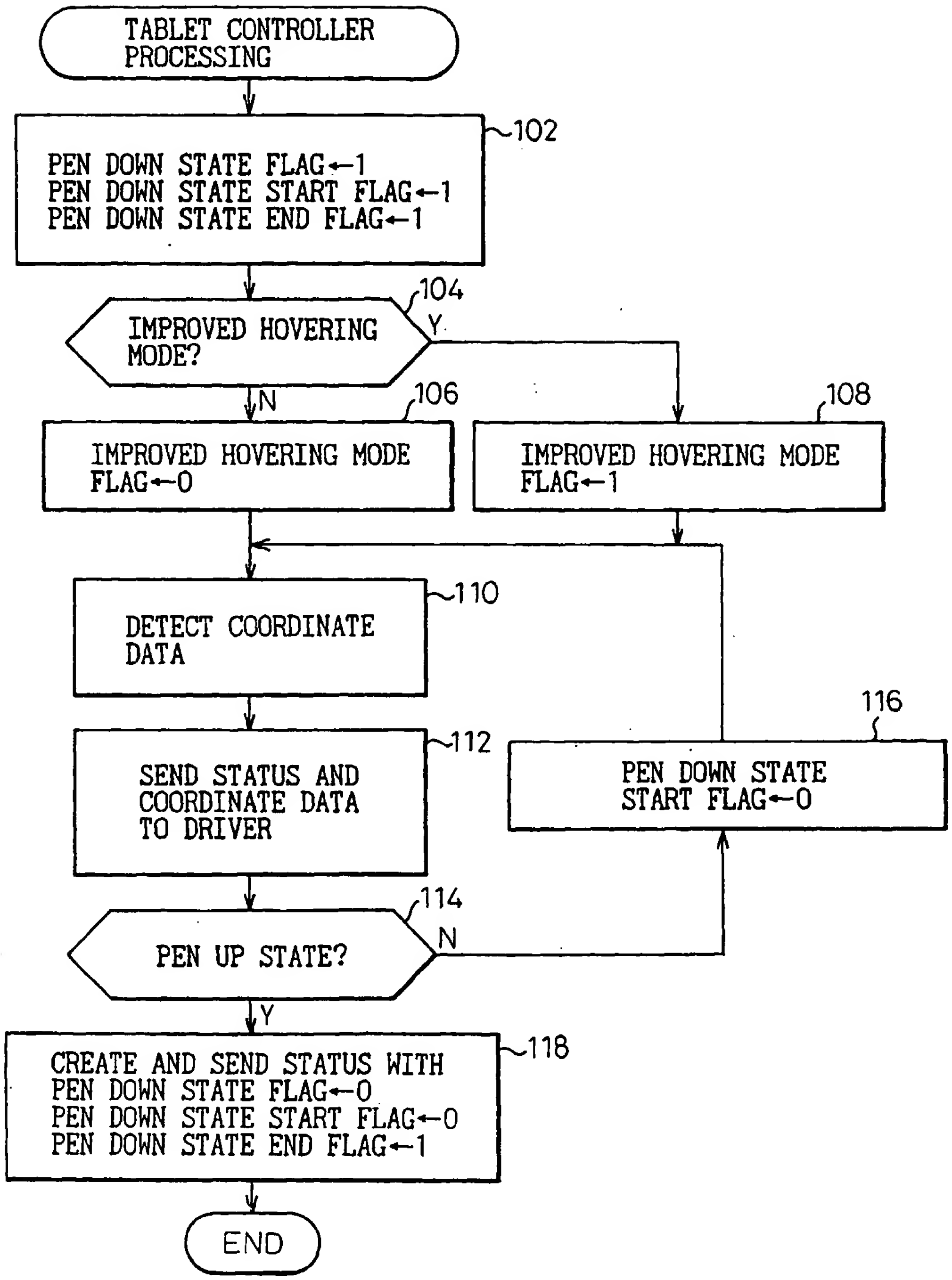

TABLET CONTROLLER PROCESSING
PEN DOWN STATE FLAG←1
PEN DOWN STATE START FLAG←1
PEN DOWN STATE END FLAG←1
102
IMPROVED HOVERING MODE?
104
Y
N
IMPROVED HOVERING MODE FLAG←0
106
IMPROVED HOVERING MODE FLAG←1
108
DETECT COORDINATE DATA
110
SEND STATUS AND COORDINATE DATA TO DRIVER
112
PEN DOWN STATE START FLAG←0
116
PEN UP STATE?
114
N
Y
CREATE AND SEND STATUS WITH
PEN DOWN STATE FLAG←0
PEN DOWN STATE START FLAG←0
PEN DOWN STATE END FLAG←1
118
END

Fig.10

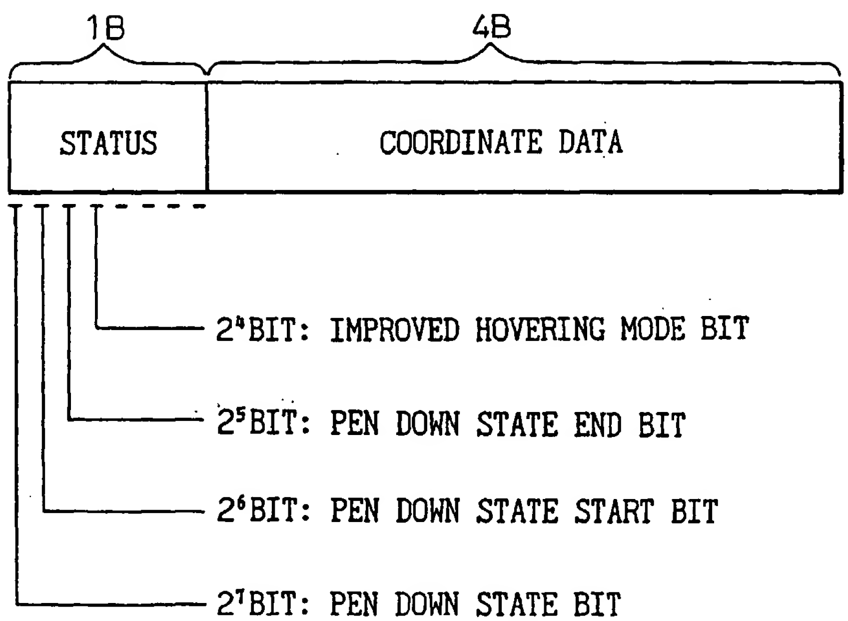

1B
4B
STATUS
COORDINATE DATA
2^4BIT: IMPROVED HOVERING MODE BIT
2^5BIT: PEN DOWN STATE END BIT
2^6BIT: PEN DOWN STATE START BIT
2^7BIT: PEN DOWN STATE BIT

Fig.11

CASE		STEPS CARRIED OUT	PEN DOWN STATE FLAG	PEN DOWN STATE START FLAG	PEN DOWN STATE END FLAG	IMPROVED HOVERING MODE FLAG
(A)	PEN DOWN STATE IS FIRST ENTERED	102,104, 108,110, 112	1	1	0	1
(B)	DURING PEN DOWN STATE	LOOP OF 114,116, 110,112	1	0	0	1
(C)	PEN UP IS DETECTED	114,118	0	0	1	1

Fig.12

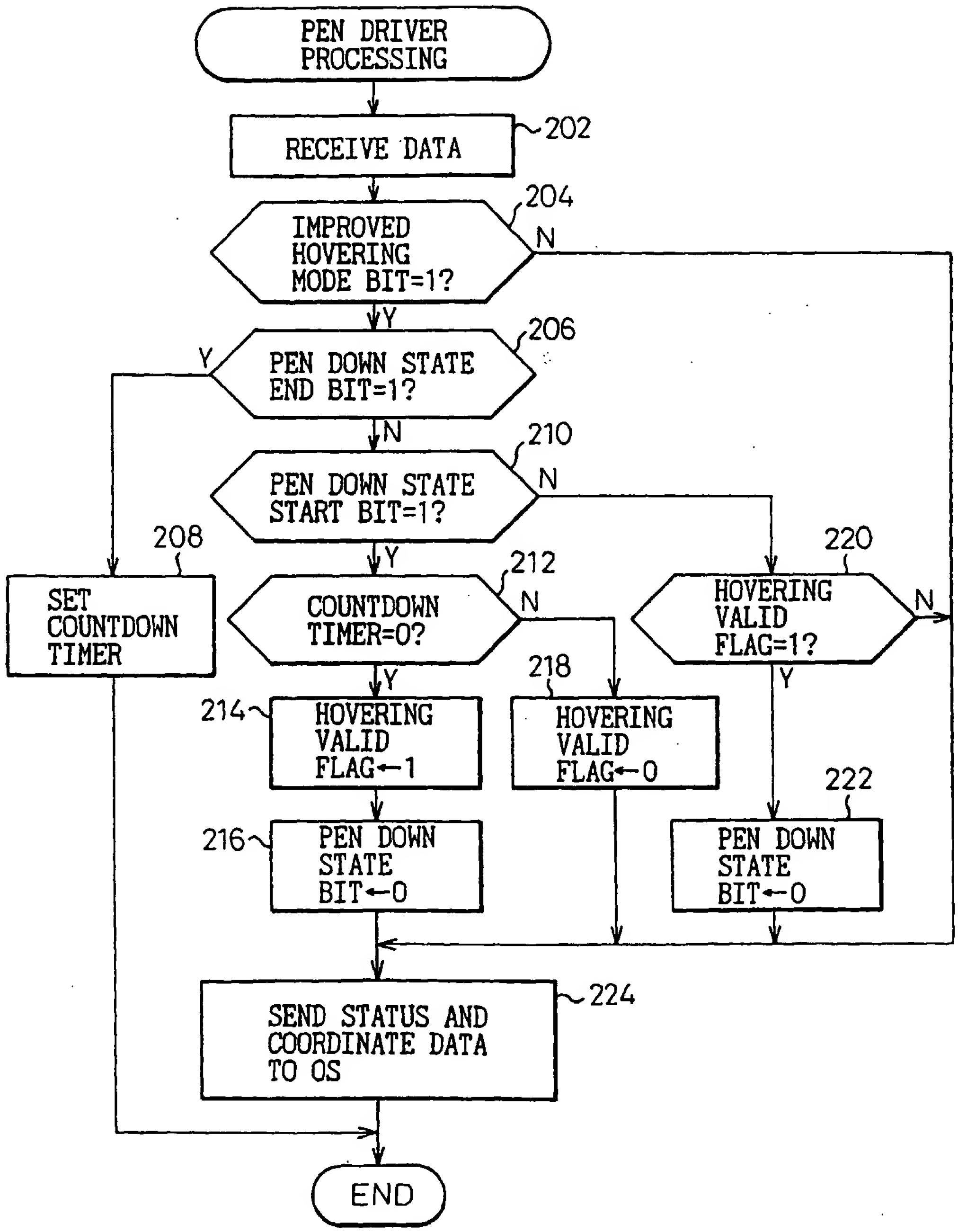

PEN DRIVER PROCESSING
RECEIVE DATA
202
IMPROVED HOVERING MODE BIT=1?
204
N
Y
PEN DOWN STATE END BIT=1?
206
Y
N
PEN DOWN STATE START BIT=1?
210
N
Y
208
SET COUNTDOWN TIMER
COUNTDOWN TIMER=0?
212
N
Y
220
HOVERING VALID FLAG=1?
N
Y
214
HOVERING VALID FLAG←1
218
HOVERING VALID FLAG←0
216
PEN DOWN STATE BIT←0
222
PEN DOWN STATE BIT←0
SEND STATUS AND COORDINATE DATA TO OS
224
END

Fig.13

CASE		PEN DOWN STATE BIT	PEN DOWN STATE START BIT	PEN DOWN STATE END BIT	HOVERING VALID FLAG	STEPS CARRIED OUT	HOVERING VALID FLAG AFTER COMPLETION OF PROCESSING	PEN DOWN STATE BIT AFTER COMPLETION OF PROCESSING
(A)	FINAL STATUS IS RECEIVED	0	0	1	–	202,204,206, 208	–	–
(B)	FIRST STATUS IS RECEIVED WITH TIMER=0	1	1	0	–	202,204,206, 210,212,214, 216,224	1	0
(C)	INTERMEDIATE STATUS IS RECEIVED FOLLOWING (B)	1	0	0	1	202,204,206, 210,220,222, 224	1	0
(D)	FIRST STATUS RECEIVED WITH TIMER≠0	1	1	0	–	202,204,206, 210,212,218, 224	0	1
(E)	INTERMEDIATE STATUS IS RECEIVED FOLLOWING (D)	1	0	0	0	202,204,206, 210,220,224,	0	1

Fig.14

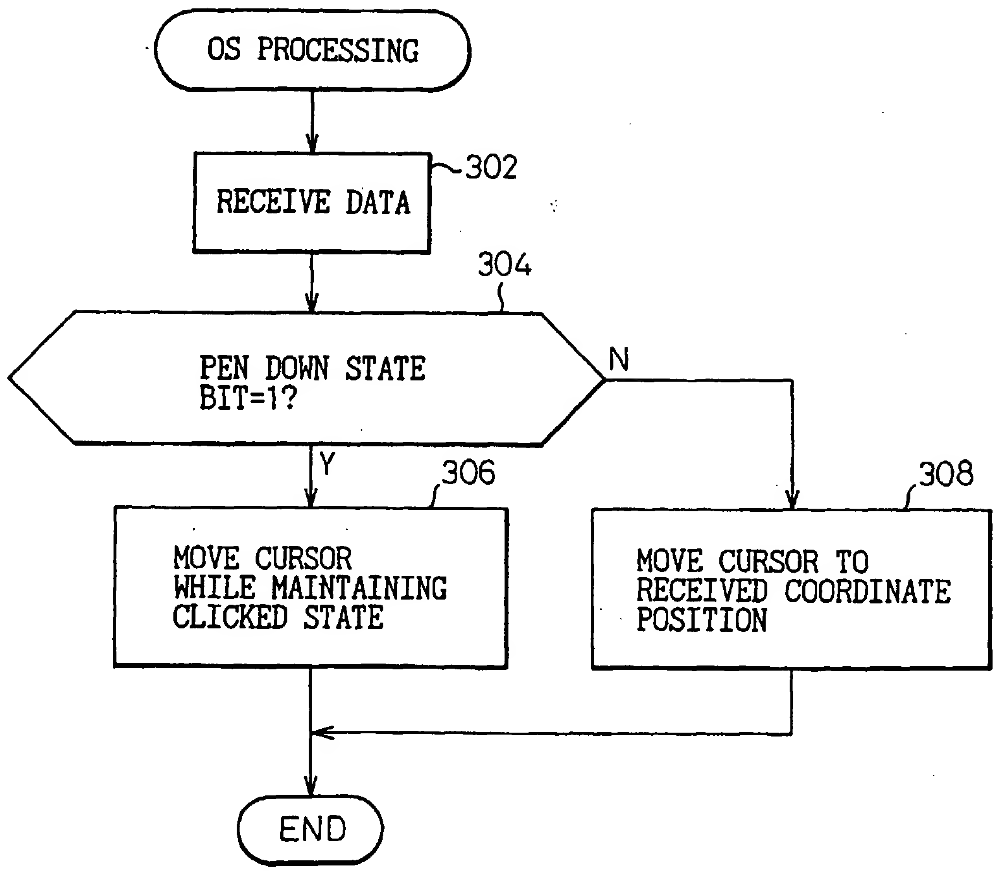

OS PROCESSING
RECEIVE DATA
302
304
PEN DOWN STATE BIT=1?
N
Y
306
308
MOVE CURSOR WHILE MAINTAINING CLICKED STATE
MOVE CURSOR TO RECEIVED COORDINATE POSITION
END

Fig.15A

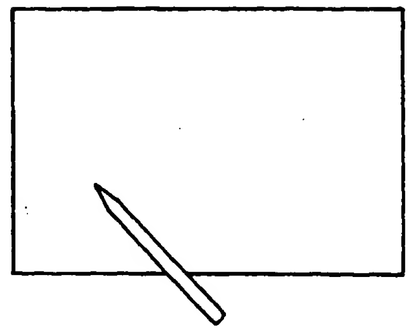


Fig.15B

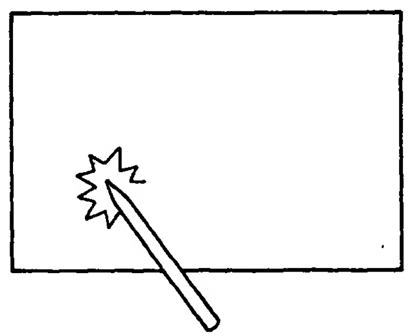


Fig.15C

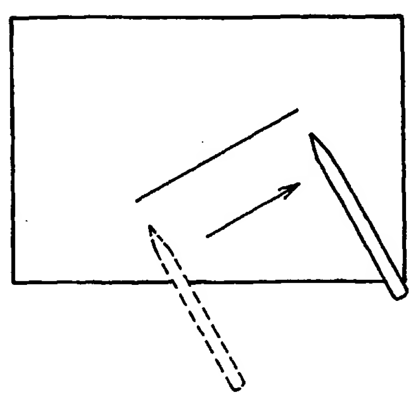


Fig.16A

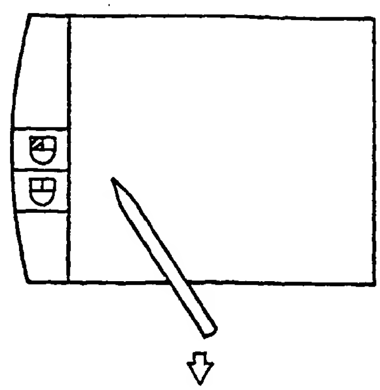


Fig.16B

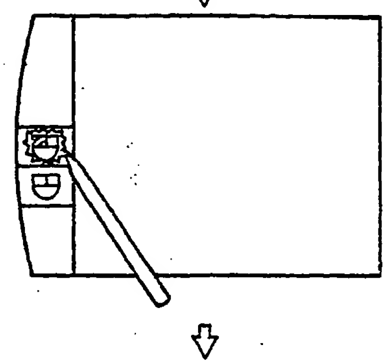


Fig.16C

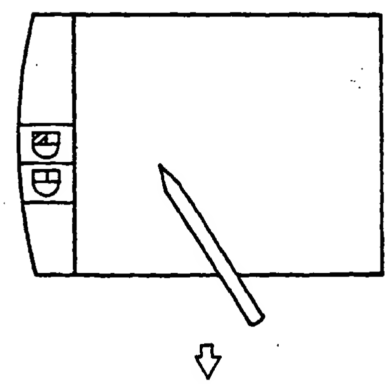


Fig.16D

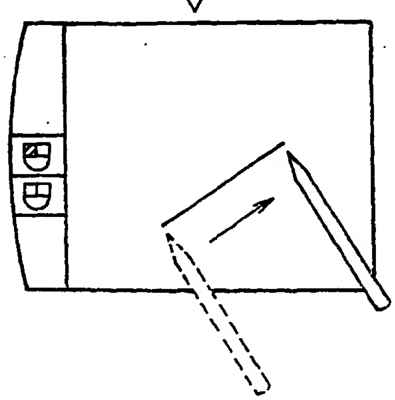


Fig.17

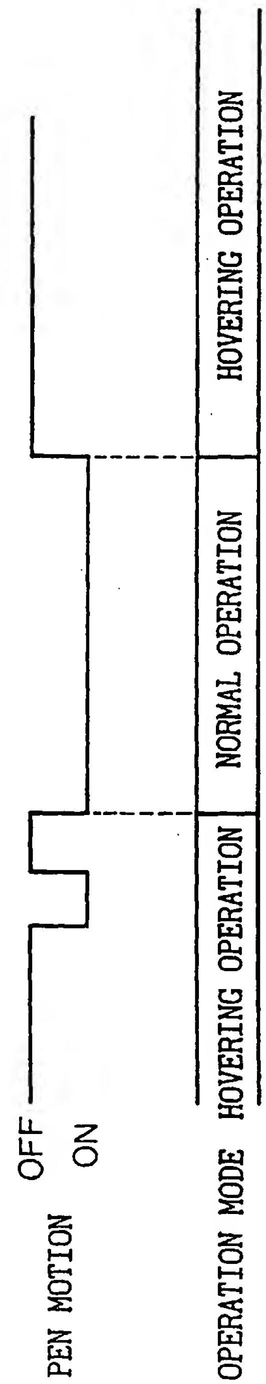

PEN MOTION
OFF
ON
OPERATION MODE
HOVERING OPERATION
NORMAL OPERATION
HOVERING OPERATION

Fig.18

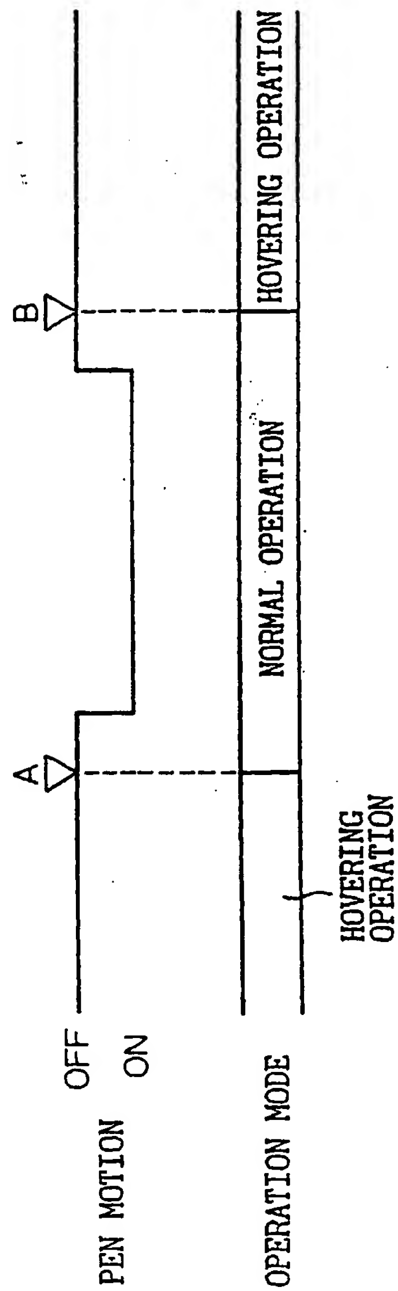

PEN MOTION
OFF
ON
OPERATION MODE
A
B
HOVERING OPERATION
NORMAL OPERATION
HOVERING OPERATION

Fig.19

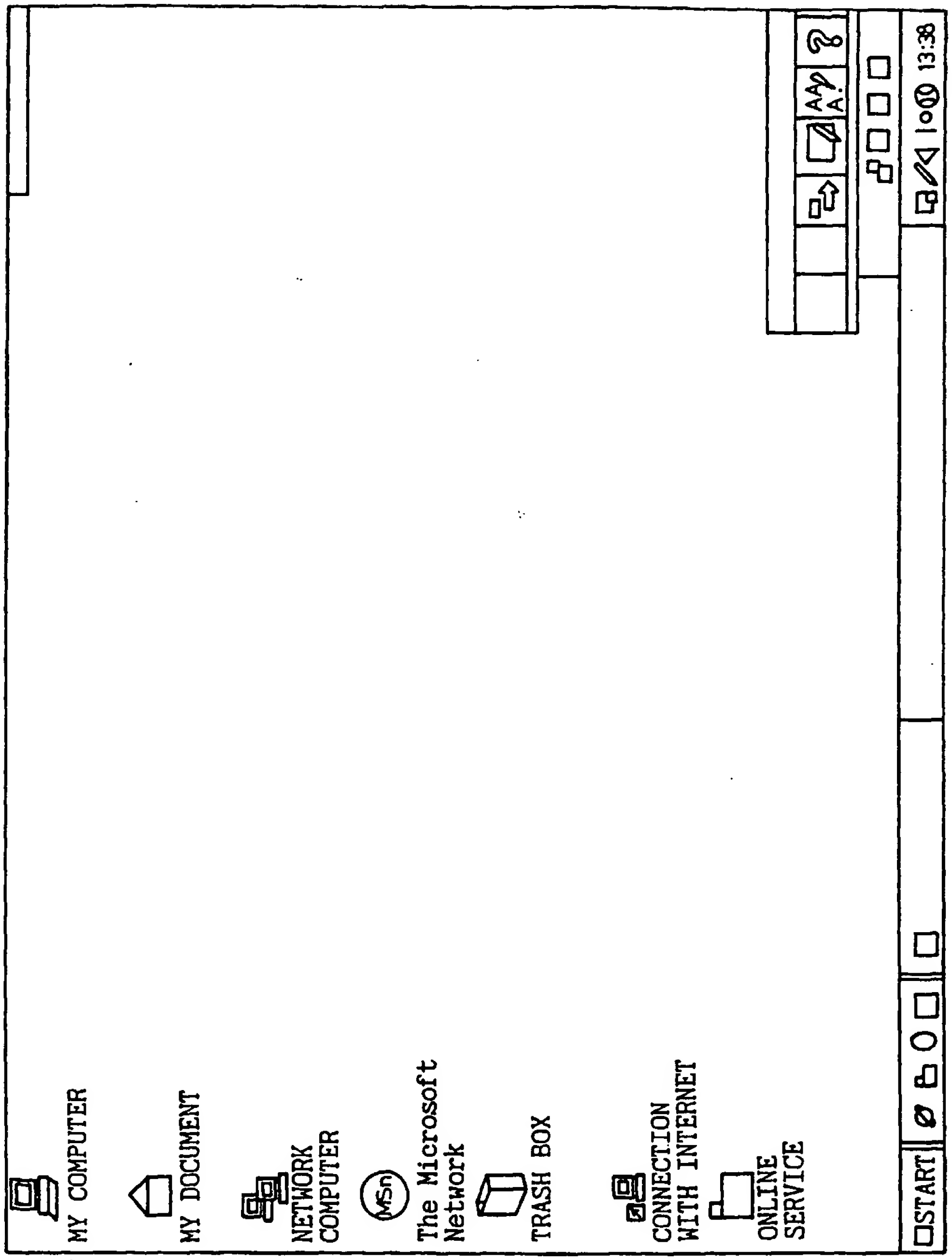

MY COMPUTER
MY DOCUMENT
NETWORK COMPUTER
The Microsoft Network
TRASH BOX
CONNECTION WITH INTERNET
ONLINE SERVICE
START
13:38

Fig.20A

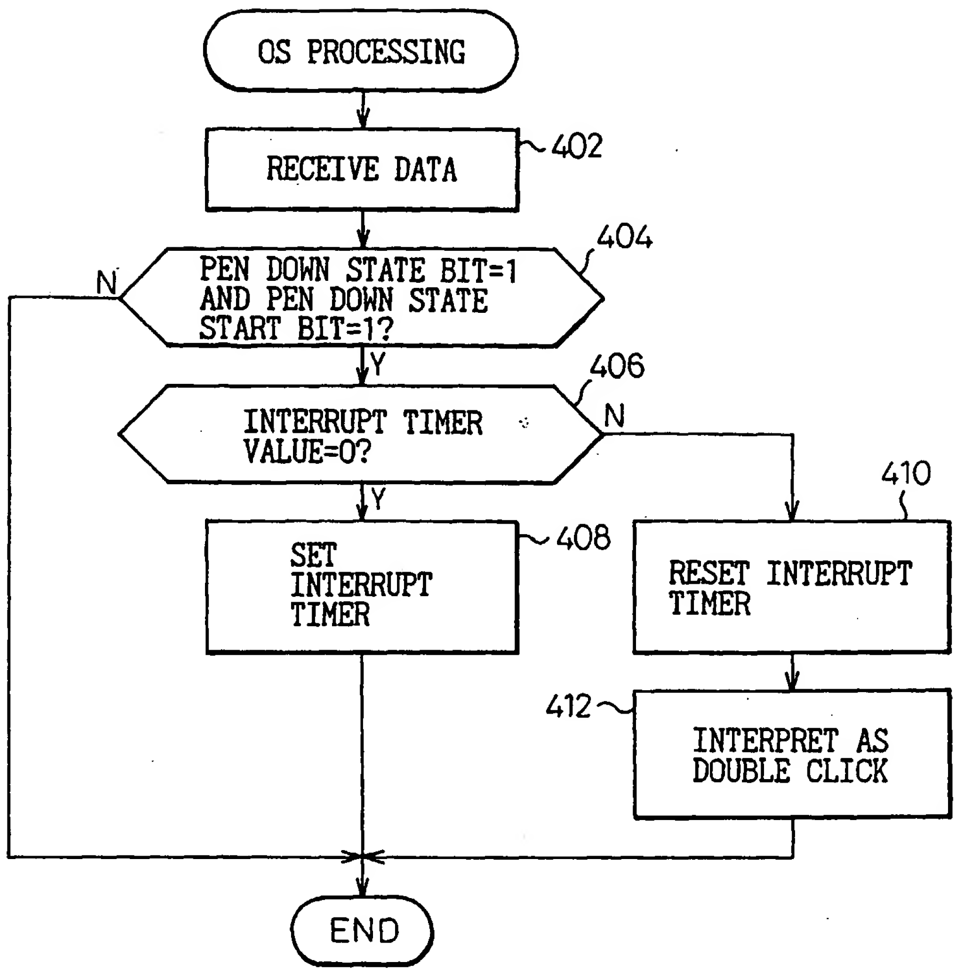


Fig.20B